

PM



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/477,101	01/04/2000	LINDEN A. DECARMO	N0003/7030	8713

21127 7590 02/12/2003

KUDIRKA & JOBSE, LLP
ONE STATE STREET
SUITE 1510
BOSTON, MA 02109

EXAMINER

ALI, SYED J

ART UNIT	PAPER NUMBER
----------	--------------

2127

DATE MAILED: 02/12/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/477,101

Applicant(s)

DECARMO, LINDEN A.

Examiner

Syed J Ali

Art Unit

2127

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☐ Responsive to communication(s) filed on ____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-16 is/are pending in the application.
- 4a) Of the above claim(s) ____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) ____ is/are allowed.
- 6) ☒ Claim(s) 1-16 is/are rejected.
- 7) ☐ Claim(s) ____ is/are objected to.
- 8) ☐ Claim(s) ____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on ____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on ____ is: a) ☐ approved b) ☐ disapproved by the Examiner.
- If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.
- 14) ☒ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
- a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892) 4) ☐ Interview Summary (PTO-413) Paper No(s). ____
- 2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948) 5) ☐ Notice of Informal Patent Application (PTO-152)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) ____ 6) ☐ Other: _____

DETAILED ACTION

Claim Rejections - 35 USC § 112

1. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 1, 7, 8, 14, and 15 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per claim 1, there are grammatical errors on lines 3 and 7 rendering the claim indefinite.

As per claim 8, there are grammatical errors on line 11 rendering the claim indefinite.

As per claim 7, there are grammatical errors on line 2 rendering the claim indefinite.

As per claim 14, there are grammatical errors on line 3 rendering the claim indefinite.

As per claim 15, there are grammatical errors on line 7 rendering the claim indefinite.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Liu et al. (USPN 5,031,089) (hereinafter Liu) in view of Achenson et al. (USPN 6,477,586) (hereinafter Achenson).

As per claim 1, Liu teaches of in a computer system, a method of distributing call flow events among a plurality of threads, each thread having an associated call flow event queue in which call flow events queued, the method comprising:

- a. determining a call flow workload level for each of the plurality of threads (col. 8 lines 8-42, “this invention uses a combination of the local queue length and the local service rate ratio at each node as the workload indicator”, wherein the workload indicator is a function of the number of jobs on the node’s queue);
- b. determining that a first of the plurality of threads is inefficiently handling its assigned call flow workload (col. 9 lines 37-63, “if the workload indicator 30 of the most busy node 12’ is above a certain high threshold, a job from the task queue 22 of that busy node 12’ is transferred to the lightly-loaded 12’”, wherein if the workload indicator is above a certain threshold it is determined that the node is inefficiently handling its call flow); and
- c. reassigning a call flow event from the call flow event queue associated the first thread to the call flow event queue associated with a second of the plurality of threads (col. 9 lines 37-63, “if the workload indicator 30 of the most busy node 12’ is above a certain high threshold, a job from the task queue 22 of that busy node 12’ is transferred to

the lightly-loaded 12", wherein the reassignment of a job from a busy node to a node which is operating under the threshold workload value is the equivalent of what is claimed).

It is noted that the method of Liu applies to nodes in a computer network redistributing jobs dynamically to obtain a balanced workload. However, the idea of redistributing tasks from queues is easily applicable to queues within any number of data structures including threads, as claimed. Further, Achenson teaches of threads that contain task queues (col. 2 line 21, "a queue associated with each of the threads"; col. 4 line 66 – col. 5 line 13, "block 30 is shown including threads and associated queues for the threads"). It would have been obvious to reconsider the invention of Liu and apply the ideology to a thread and its associated queue rather than a computer node since in a case where a balanced workload is sought in a single multithreaded system rather than a multinode network the desired effect is equivalent. In this manner, a balanced workload is achieved and allows for an increase in the efficiency of a thread's operation as well as ensuring that all threads are doing an equivalent amount of work.

As per claim 2, Liu teaches the method according to claim 1 further comprising the step:

d. processing the call flow events associated with each of the plurality of threads (This is inherent in Liu in the sense that the method described therein is in reference to tasks scheduled for execution in a computer network. The invention therein is more concerned with how jobs are transferred between nodes, and the execution of those jobs once a balance has been achieved is not specifically stated, but understood to be an inherent step).

As per claim 3, Liu teaches the method according to claim 1 wherein step c further comprises:

- c1. removing a call flow event from the call flow event queue associated within the first thread (col. 7 lines 12-41, “logic for transferring a job from the queue of the other of the nodes having the highest workload value over a pre-established value ... to the queue of the checking node”, wherein when the job is transferred out of the node having the highest workload value over a pre-established value, the job is also removed from that node’s task queue); and
- c2. placing the removed call flow event in the call flow event queue associated with the second thread (col. 7 lines 12-41, “logic for transferring a job from the queue of the other of the nodes having the highest workload value over a pre-established value ... to the queue of the checking node”, wherein when the job is transferred, the job is added to the task queue of the receiving node which was determined to have a light burden).

As per claim 4, Liu teaches the method according to claim 1 wherein step c further comprises:

- c1. selecting the second thread in accordance with the number of call flow events in the call flow event queue associated with the second thread (col. 9 lines 13-36, “If the workload indicator 30 is less than a certain low threshold level, the lightly-loaded node 12’ initiates a search for the most busy node”, wherein the selection of the second thread, i.e. the thread which is to receive tasks from a burdened thread, is selected based on the

fact that the workload indicator shows that the node is lightly-loaded and could handle an increase in its allocated tasks).

As per claim 5, Liu does not specifically teach the method according to claim 1 wherein step c further comprises:

- c1. allocating the call flow events to a thread within the computer system with the least call flow load.

However, Liu teaches of allocating tasks to a node within the computer system with a workload under a threshold value, and taking tasks from the queue of the most heavily burdened node. In essence, this achieves the same result as a task is taken from a node that is known to be overloaded and attempted to move to a node that is known to be operating under a workload that is lighter than it can handle. However, this issue is addressed in Liu (col. 8 lines 54-61, "the danger of sender-initiated schemes, i.e. that all the nodes are overloaded and each attempts to give away jobs causing unproductive overhead which simply further saturates the already overloaded system, is eliminated"). Therefore, Liu directly addresses this issue and decided that less overhead would be created by transferring to a lightly-loaded node a job from a known to be overloaded node, rather than vice versa. Certainly, a sender-initiated scheme has its advantages as well, but this is an obvious modification and does not stand as distinct from the prior art disclosed in Liu.

As per claim 6, Liu teaches the method according to claim 1 wherein step b comprises:

b1. determining whether the number of call flow events in the call flow event queue associated with a thread has exceeded a predetermined criteria (col. 8 line 65 – col. 9 line 12, wherein a rudimentary algorithm is shown that calculates the workload as queue length divided by service rate. Queue length is earlier defined in Liu as being the number of tasks remaining to be serviced by the node. Subsequently, a determination is made if the workload value, as defined above, is above a pre-determined value. This occurs if the number of tasks to be processed by a node exceeds a predetermined criteria depending on its service rate.)

As per claim 7, Liu teaches the method according to claim 1 wherein step a comprises:

a1. assigning call flow events among the call flow event queues associated the respective plurality of threads in the system (fig. 1 elements 18, 20, wherein Liu discloses prior art that shows how a control computer initially distributes tasks among a plurality of available processing nodes. Additionally, “Official Notice” is taken that it is well known in the art that there are a great number of scheduling algorithms that will distribute tasks among a number of available processing nodes, and distribute them in a manner that balances the load equally among all nodes.)

As per claims 8-14, the method of Liu is intended for use in a computer program product for use with a computer system, the computer system operatively coupled to a computer network and capable of communicating with one or more processes over the network, the computer program product comprising a computer usable medium having program code embodied in the

medium (fig. 4, wherein a basic network for implementing the method of Liu is disclosed, and it is clearly shown that it is intended for use as part of a computer system coupled to a network. Further, basic code is described in col. 8 line 65 – col. 9 line 12 that indicates that the method disclosed therein is intended for use in a computer usable medium having program code embodied in the medium).

Additionally, the remainder of the limitations in the claims 8-14 reflect those of claims 1-7. Claims 8-14 are therefore rejected for similar reasons as stated for claims 1-7, respectively.

As per claim 15, Liu teaches of in a computer system, an apparatus for distributing call flow events among a plurality of threads, each thread having an associated call flow event queue in which call flow events queued, the apparatus comprising:

a call flow engine configured execute call flow events associated with one of the threads (fig. 3, wherein Liu discloses a connection via dotted line to the operating system indicating execution of the jobs in a node's queue once a sufficient balance has been achieved);

a call flow manager configured to distribute a plurality of call flow events among a plurality of threads used for managing the processing of plurality of call flows (fig. 3 element 28, wherein the "task allocation and transfer logic" allocates jobs to a specific node and attempts to achieve a balanced processing between all nodes of the system), n the call flow manager optimizing the processing of the call flows by determining which plurality of threads are operating inefficiently and reassigning a portion of the call flow events assigned to the inefficient thread to other of the plurality of threads having excess

call flow processing capacity (col. 9 lines 37-63, "If the workload indicator 30 of the most busy node 12' is above a certain high threshold, a job from the task queue 22 of that busy node 12' is transferred to the lightly-loaded node, wherein the transfer logic determines if a node has excess processing capability, and if so determines which node is operating at the highest burden, and transfers jobs out of the heavily burdened node and into the node with excess processing capability).

It is noted that the method of job transferring discussed in Liu is in reference to nodes of a computer network rather than threads of a multithreaded system. However, Achenson discusses job queues associated with a particular thread, as discussed for claim 1. The discussion of claim 1 provides the motivation for one of ordinary skill in the art to combine Liu and Achenson so that the method of Liu could be applied to a multithreaded system where each thread has a task queue as well.

As per claim 16, Liu teaches the apparatus of claim 15 wherein the call flow manager continues to reassign call flow events until a balanced call flow event processing level is attained among the plurality of threads (col. 8 lines 46-63, wherein Liu discusses that the method described therein is meant to dynamically balance the workload, and as such adapts to changes in the workload of the system, i.e. if nodes are known to be operating at less than capacity, a search is made to determine if other nodes require assistance, as well as if a node is idle, searching similarly. This dynamic approach serves to balance the call flow as it changes until a satisfactory balance is attained).

Art Unit: 2127

Conclusion

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 1st Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John A Follansbee can be reached on (703) 305-8498. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.



Syed Ali
February 5, 2003



MAHIB BANANKHAH
PRIMARY EXAMINER